# Gossiping correspondences to reduce semantic heterogeneity of unstructured P2P systems

Thomas Cerqueus, Sylvie Cazalens, Philippe Lamarre

# Gossiping correspondences to reduce semantic heterogeneity of unstructured P2P systems

Thomas Cerqueus, Sylvie Cazalens and Philippe Lamarre

LINA, University of Nantes
{thomas.cerqueus, sylvie.cazalens, philippe.lamarre}@univ-nantes.fr

**Abstract.** In this paper we consider P2P data sharing systems in which each participant uses an ontology to represent its data. If all the participants do not use the same ontology, the system is said to be semantically heterogeneous. This situation of heterogeneity prevents perfect interoperability. Indeed participants could be unable to treat queries for which they do not understand some concepts. Intuitively, the more heterogeneous a system, the harder to communicate. We first define several measures to characterize the semantic heterogeneity of P2P systems according to different facets. Then, we propose a solution, called CorDis, to reduce the heterogeneity by decreasing the gap between peers. The idea is to gossip correspondences through the system so that peers become less disparate from each other. The experiments use the PeerSim simulator and ontologies from OntoFarm. The results show that CorDis significantly reduces some facets of semantic heterogeneity while the network traffic and the storage space are bounded.

## 1 Introduction

We consider peer-to-peer (P2P) data sharing systems where semantic meta-data are used to represent information and to enhance search. This general setting can be instanciated in different ways depending on the kind of meta-data used. We focus on applications where each peer uses an ontology to represent the information it stores. Typical examples are indexing documents or data sets with respect to the concepts of the ontology, or annotating the elements of a database schema with entities of the ontology.

The use of different ontologies results in semantic heterogeneity of the system. Because some peers are unable to precisely understand each others, some semantic interoperability has to be reached in some way. It is generally assumed that neighbour peers use alignments between their ontologies [7]. Then, knowing correspondences between entities of ontologies, each peer translates incoming queries before forwarding them. This kind of approach works well in some cases, although it suffers from information losses due to several translations [8, 3]. Moreover, it mainly focuses on leveraging interoperability without considering reducing semantic heterogeneity. Our goal is to define a class of algorithms that reduce the semantic heterogeneity of the P2P system, thus leveraging interoperability as a consequence. We proceed in two steps.

The first step consists in characterizing semantic heterogeneity. Apart from some intuitions like "the more different ontologies are used in the system, the higher heterogeneity is", or "the more alignments are known, the lower heterogeneity is", no definition of semantic heterogeneity exists (at least to our knowledge). Based on the observation that the concept of heterogeneity has several dimensions (or facets), we propose several definitions to capture them.

The goal of the second step is to define algorithms that make semantic heterogeneity decrease along some dimensions. Of course, a simple way to decrease heterogeneity is having the peers use exactly the same ontology. We believe that this is not realistic when peers are numerous and with different backgrounds. Hence we focus on solutions that have the peers increase their knowledge of alignments. Assuming that they join the system with already known alignments, the probability that all of them know exactly the same alignments is very low. Thus the idea is to make the peers share their knowledge by *disseminating* correspondences between entities of different ontologies. We consider a case where peers trust each others: no correspondence should be disregarded because it has been forwarded by an untrusted peer.

In order to implement dissemination of correspondences we use a *gossiping* algorithm in the sense of [11]: each peer regularly picks up some other peer for a two-way information exchange. In our case, each peer selects some correspondences to send to another peer. This latter also selects correspondences and send them to the former. After several rounds correspondences disseminate across the system. The CorDis protocol is based on this idea. In addition, because peers generally have limited local storage, a scoring function is used to order the correspondences and store the most relevant ones. Relevance is computed considering a history of the incoming queries. We propose to favour the correspondences that involve entities that appeared in recent queries, and, to some extent chosen by the programmer, those involving entities belonging to ontologies referred to in recent queries. The scores of the correspondences are regularly updated, so that the CorDis protocol adapts the information exchange to the current queries.

In this paper, we bring several contributions. After presenting our formal model (section 2), we first propose several definitions of semantic heterogeneity measures, corresponding to different facets of this notion (section 3). Second, we propose the CorDis gossip-based protocol to disseminate correspondences across the system (section 4). It considers a history of queries to score the correspondences. Thus it ensures some flexibility with respect to current queries. Third, we report on several experiments conducted with the PeerSim simulator and fifteen ontologies from OntoFarm (section 5). The CorDis protocol is evaluated with respect to the proposed measures of semantic heterogeneity. The results show that CorDis significantly reduces several facets of heterogeneity while the network traffic and the storage space are bounded. This work builds on previous results concerning ontology mapping, ontology distances and gossiping algorithms. However, it does not have any equivalent among the previously proposed solutions to improve semantic interoperability (section 6).

## 2 Hypothesis and model

### 2.1 The P2P system

We assume that each peer $p$ has a unique identifier, denoted by $id(p)$. To ensure relationships with other peers, peer $p$ maintains a routing table $table(p)$, composed of a set of peer identifiers which are called $p$'s neighbours.

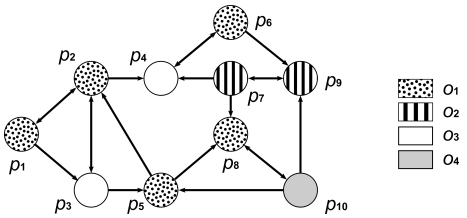


**Fig. 1.** Unstructured P2P system.

**Definition 1 (Unstructured P2P system)** *An unstructured P2P system is defined by a graph $\mathcal{S} = \langle \mathcal{P}, \mathcal{N} \rangle$, where $\mathcal{P}$ is a set of peers and $\mathcal{N}$ represents a neighbourhood relation defined by: $\mathcal{N} = \{(p_i, p_j) \in \mathcal{P}^2 : p_j \in table(p_i)\}$.*

In the system presented in Fig. 1 the neighbourhood of $p_1$ within a radius equal to 2 is composed of $p_2$, $p_3$, $p_4$ and $p_5$.

### 2.2 Ontologies and alignments

We consider that an ontology is composed of a set of concepts $C_o$, a set of relations $R_o$ (linking concepts) and a set of properties $P_o$ (assigned to concepts). The union of these three sets of entities is denoted by $E_o$. In practice OWL [13] allows to represent ontologies by defining *classes*, *datatype properties* and *object properties*. We assume that each ontology is uniquely identified by an URI. Thus two ontologies are equal if and only if their URIs are the same. We assume that a peer uses the same ontology during its life-time.

An alignment process aims at identifying a set of correspondences between the entities of two ontologies [7].

**Definition 2 (Correspondence)** *A correspondence is a 4-tuple $\langle e, e', r, n \rangle$ such that $e$ (resp. $e'$) is an entity from $o$ (resp. $o'$), $r$ is a relation between $e$ and $e'$, and $n$ is a confidence value.*

An alignment between ontologies of Fig. 2 could contain the correspondences: $\langle Thing_1, Thing_2, \equiv, 1 \rangle$, $\langle Flower_1, Flower_2, \equiv, 1 \rangle$, $\langle odour_1, fragrance_2, \equiv, 1 \rangle$ and $\langle Edelweiss_1, Flower_2, isA, 1 \rangle$. Notice that an alignment is not necessarly perfect in the sense that some correct correspondences may be missing and others may be incorrect. Here, we assume that an alignment does not contain incorrect correspondences.

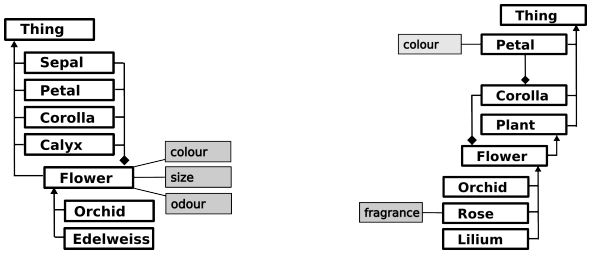


**Fig. 2.** Two ontologies $o_1$ and $o_2$ composed of concepts, properties and relations.

**Definition 3 (Peer-to-ontology mapping)** *Given a P2P system* $\mathcal{S} = \langle \mathcal{P}, \mathcal{N} \rangle$ *and a set of ontologies* $\mathcal{O}$*, a peer-to-ontology mapping is a function* $\mu : \mathcal{P} \rightarrow \mathcal{O}$*, mapping each peer to one ontology.*

In order to understand incoming queries each peer must know correspondences. We denote by $\kappa_p$ the set of correspondences stored by a peer $p$ and $\kappa_p(o, o')$ denotes the subset of $\kappa_p$ concerning ontologies $o$ and $o'$.

### 2.3 Disparity between two peers

We introduce the notion of disparity function to quantify the difference between two peers.

**Definition 4** *A disparity function* $d : \mathcal{P} \times \mathcal{P} \rightarrow [0, 1]$ *is a function that assigns a real value in* $[0, 1]$ *to a couple* $\langle p, p' \rangle$ *representing how much* $p'$ *differs from* $p$*. It satisfies the minimality property:* $\forall p \in \mathcal{P}, d(p, p) = 0$*, but we do not assume it is a mathematical distance.*

There are different ways to define disparity and several proposals exist [12, 5]. Some consider the alignments between the peers' ontologies [5].

### 2.4 Semantic heterogeneity of a system

The following definition states what a semantic heterogeneity function is. It does not mean that heterogeneity can be captured by a single measure. Rather, depending on the application, several complementary measures could be used.

**Definition 5** *Let* $\mathcal{SM}$ *be a set of models* $\mathcal{M} = \langle \mathcal{S}, \mathcal{O}, \mu, d \rangle$ *where* $\mathcal{S}$ *is a P2P system,* $\mathcal{O}$ *is a set of ontologies,* $\mu$ *is a peer-to-ontology mapping, and* $d$ *is a disparity function between peers.*
*A semantic heterogeneity measure is a function* $\mathcal{H} : \mathcal{SM} \rightarrow [0, 1]$ *such that:*

- $\mathcal{H}(\mathcal{M}) = 0$ *if* $|\{o \in \mathcal{O} : \exists p \in \mathcal{P} \text{ s.t. } \mu(p) = o\}| = 1$ *(minimality);*
- $\mathcal{H}(\mathcal{M}) = 1$ *if* $\forall p \neq p' \in \mathcal{P}, d(p, p') = 1$ *(maximality).*

The conditions express that (i) homogeneity occurs when the same ontology is used by all the peers and that (ii) maximal heterogeneity occurs when all the disparities between peers are maximal.

## 3 Semantic heterogeneity measures

In this section, we propose measures which are general enough to be used in many application domains while still being meaningful.

### 3.1 Disparity unaware measure

Notion of diversity is commonly used to measure the heterogeneity of a population (*e.g.* in biology). Richness partly characterizes the diversity of a population. In our context it depends on the number of different ontologies used in the system. If all the peers use the same ontology, then the system is completely homogeneous. By cons, the more ontologies there are, the more heterogeneous it is. This idea can be expressed by the following measure:

$$\mathcal{H}_{Rich}(\mathcal{M}) = \frac{|o_{\mathcal{S}}| - 1}{|\mathcal{P}| - 1}$$

where $|o_{\mathcal{S}}|$ is the number of different ontologies used in the system $\mathcal{S}$, and $|\mathcal{P}|$ the number of peers. In the system presented on Figure 1, four different ontologies are used by the ten participants: $\mathcal{H}_{Rich}(\mathcal{M}) = \frac{4-1}{10-1} = 0.33$. Measuring richness allows to draw preliminary conclusions. In particular it gives information about the need of alignments to reach interoperability. A richness value equal to 0 means that heterogeneity is null: no alignment is needed to ensure interoperability in the system. A value equal to 1 means that heterogeneity is total: alignments are needed between each pair of participants to communicate.

### 3.2 Disparity aware measures

**Topology unaware measure** We propose to consider disparity between peers rather than only consider the ontologies they use. If the disparity between peers is globally important, it means that peers have important knowledge differences. The more different their knowledge, the harder to communicate (*i.e.* answering queries). Indeed an important loss of information will occur during query translation. As we do not take into account the system topology, we consider the disparity between each pair of peers:

$$\mathcal{H}_{Disp}(\mathcal{M}) = \frac{1}{|\mathcal{P}|^2 - |\mathcal{P}|} \sum_{p_i \neq p_j \in \mathcal{P}} d(p_i, p_j)$$

The $\mathcal{H}_{Disp}$ measure determines if peers are globally disparate from each other.

**Topology aware measure** We propose to take into account how disparate peers are with regards to their neighbourhoods. If peers are globally far (semantically speaking) from their respective neighbourhoods, the system is highly heterogeneous. Contrariwise, if peers are close to their neighbourhoods, the system is weakly heterogeneous, even if the diversity of the system is not null.

We denote by $\mathcal{N}_r(p)$ the neighbourhood of a peer $p$ within a radius $r$. It is the set of peers accessible from $p$ with $l$ hops, where $1 \leqslant l \leqslant r$. We consider

that $p$ does not belong to $\mathcal{N}_r(p)$. We first propose a measure that focuses on a given peer and determines how this latter is understood by its neighbours:

$$\mathcal{H}^r_{Dap}(\mathcal{M}, p) = \frac{1}{|\mathcal{N}_r(p)|} \sum_{p_i \in \mathcal{N}_r(p)} d(p, p_i)$$

A global measure can be obtained:

$$\mathcal{H}^r_{DapAvg}(\mathcal{M}) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathcal{H}^r_{Dap}(\mathcal{M}, p)$$

If $\mathcal{H}_{DapAvg}$'s value is weak, it means that peers are globally close to their neighbours: each peer is surrounded by peers able to "understand" it.

**Proposition 1.** *All the measures introduced in this section satisfy both properties of minimality and maximality (proof is trivial).*

## 4 Gossiping correspondences

### 4.1 Principles of gossip-based protocols

Our approach is based on a gossip-based protocol that disseminates data [11]. In such a protocol, each peer consists of two threads: an active and a passive one. The active thread is used to initiate communications with another peer. We assume that the peer selection is ensured by a peer sampling service, allowing peers to uniformaly and randomly select another peer [9]. Thus, each peer regularly contacts another peer to exchange information. We consider that the size of a message does not exceed $m_{max}$. When a peer is contacted by another one (through the passive thread), the former has to answer by sending some information. Thus, both peers treat the received information. This principle is explicited by algorithms 1 and 2. In these algorithms, peers have to process two crucial tasks: data selection and data processing.

**Algorithm 1**: Active thread: $p_0$

```
1 while true do
2     p_1 ← selectPeer() ;
3     localBuff ← selectToSend() ;
4     sendTo(p_1, localBuff) ;
5     receiveFrom(&p_1, &remoteBuff) ;
6     processData(remoteBuff) ;
7     wait(T) ;
```

**Algorithm 2**: Passive thread: $p_1$

```
1 receiveFromAny(&p_0, &remoteBuff) ;
2 localBuff ← selectToSend() ;
3 sendTo(p_0, localBuff) ;
4 processData(remoteBuff) ;
```

### 4.2 The CorDis protocol

The main idea of this protocol is to disseminate information over the network to share correspondences known by some but ignored by others in order to reduce some facets of semantic heterogeneity of the system. In the remaining of this

work, we do not make any assumption about the way queries are transmetted in the system, but we consider that they are unchanged during the propagation: each peer receives the same query, and is responsible to translate it if necessary.

When the process starts, each peer $p$ knows some correspondences, a subset of which involves its own ontology (noted $init_p$). This subset of $o_p$-correspondences (correspondences that involve its own ontology) should always be recorded by the peer. The purpose of dissemination is that each peer learns additional correspondences that might be useful to it to translate the queries it receives into its own ontology. We disseminate the correspondences by gossiping: Each peer $p$ regurlarly initiates an exchange of correspondences with another peer $p'$. It selects some correspondences it knows and sends them to $p'$. In turn, $p'$ chooses among the correspondences it stores and send them to $p$.

**Storage of correspondences** Each peer must store the correspondences it has been informed of in some cache, of limited size, thus preventing the peer from storing all the correspondences. Choice of the correspondences to keep is obtained by a scoring function which enables to order the correspondences: only the best ones are kept. In theory, the scoring function could be specific to each peer. Here we propose that each of them consider a history of the received queries.

A **history** of received queries is made of two lists $\mathcal{L}_1$ and $\mathcal{L}_2$. List $\mathcal{L}_1$ contains the entities used in the last $k$ received queries, while $\mathcal{L}_2$ contains the ontologies used to express the last $k'$ received queries. Notice that an item can appear several time in a list if it has been involved in several queries. The intuition of the scoring function is that peers favour the correspondences that might be useful for translating queries (it can be useful locally, or for others).

**Definition 6 (Scoring function)** *Given a set of correspondences $\mathcal{C}$, we define the scoring function of a peer $sc : \mathcal{C} \to [0, 1]$ as:*
$sc(\langle e, e' \rangle) = \omega \cdot [f_1(e) + f_1(e')] + (1 - \omega) \cdot [f_2(o) + f_2(o')]$ *where $e \in o$, $e' \in o'$, and $f_1$ (resp. $f_2$) measures the frequency of occurence of an element in $\mathcal{L}_1$ (resp. $\mathcal{L}_2$).*

The coefficient $\omega \in [0, 1]$ is used for giving more or less importance to a correspondence involving entities that do not appear in recent queries, but that belong to ontologies used recently. If the focus of interest of the queries changes, the scoring values of the correspondences will change, giving more importance to relevant correspondences. Scores are regularly calculated to take dynamicity into account.

Because the correspondences involving its own ontology are of prime importance for the peer, we propose that it tries to store as much possible of them (or all of them if possible) in a specific **repository**, including $init_p$, distinct from the cache which is then devoted to the other correspondences. If the repository is too small for storing all the $o_p$-correspondences, the peer can use the scoring function to eliminate some of them. We denote by $repository(p)$ the repository of a peer $p$, and by $cache(p)$ its cache (respectively limited to $r_{max}$ and $c_{max}$ entries).

**Data selection** When a peer has to send correspondences, it selects them from both the cache and the repository. We introduce the number $\pi \in [0, 1]$ to repre-

sent the ratio of correspondences to select in both sets. Thus, a peer randomly selects $[\pi \cdot m_{max}]$ correspondences in its repository, and $[(1-\pi) \cdot m_{max}]$ in its cache. Fig. 3 summarizes this process. Random selection is used to ensure that two correspondences of the repository (resp. the cache) have the same probability to be sent.

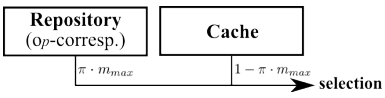


**Fig. 3.** Data selection process.

**Data processing** When a peer $p$ receives a message, it executes two main tasks. First it computes the score of the correspondences in $msg$ and then merges them with its local data. It only consists in adding $o_p$-correspondences in $repository(p)$ and the others in $cache(p)$ and re-order the correspondences. If a correspondence is already stored, the newest score is used. Then, the best $r_{max}$ (resp. $c_{max}$) correspondences are kept in the repository (resp. in the cache).

## 5 Preliminary experiments

In this section we study the performances of our protocol w.r.t. application parameters, initial heterogeneity, and dynamicity of the system.

We used the PeerSim simulator [10] to generate P2P systems as directed graphs. In order to simulate real-world situations we use the OntoFarm dataset [16, 5]. It is composed of fifteen ontologies, expressed in OWL, dealing with the conference organization domain. Ontologies are composed of 51 concepts in average (between 14 and 141) and their average volume is 41.3 KB (between 7.2 KB and 100.7 KB). We use a Poisson law to distribute ontologies in the system. Thus some ontologies are more used than others. We consider it as a realistic situation. As we only have fifteen ontologies, we consider relatively small systems (*i.e.* with 100 peers) to ensure a sufficient degree of heterogeneity. Moreover each peer has three other peers as neighbours.

We set $\pi = 0.5$ so that correspondences are fairly kept from the repository and the cache. Furthermore we consider that histories constantly change over time: scoring function values vary continually. It is considered as a critical situation.

We exploit the alignments used in [5] as reference alignments between ontologies. In average 98 correspondences are available from one ontology to others (altogether 1470 correspondences). As each correspondence is an equivalence between two concepts (with $n = 1$), we adapt the coverage measure presented in [5] as the measure of disparity between two peers. It is defined as:

$$d(p,p') = \frac{|\{e \in E_o : \forall e' \in E_{o'}, \nexists \langle e, e', \equiv, 1\rangle \in \kappa_{p'}(o,o')\}|}{|E_o|}$$

where $o$ and $o'$ are the ontologies of $p$ and $p'$, and $\kappa_{p'}(o,o')$ is the set of correspondences that $p'$ knows between $o$ and $o'$. This definition expresses how $p'$ can understand $p$'s queries.

| | $init_p$ | $r_{max}$ | $c_{max}$ | $m_{max}$ | LS (KB) | NT (KB) |
|---|---|---|---|---|---|---|
| $c_{ref}$ | 25 | 75 | 20 | 20 | 20.9 | 4.4 |
| $c_1$ | 5 | 75 | 20 | 20 | 20.9 | 4.4 |
| $c_2$ | 25 | 150 | 20 | 20 | 37.4 | 4.4 |
| $c_3$ | 25 | 75 | 75 | 20 | 33.0 | 4.4 |
| $c_4$ | 25 | 75 | 20 | 10 | 20.9 | 2.2 |

**Tab. 4.** Configurations studied in section 5.1, and theorical analysis of the local storage (LS) per peer, and the network traffic (NT) per cycle.

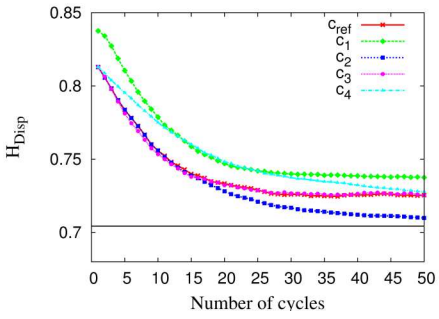


**Fig. 5.** Decrease of $\mathcal{H}_{Disp}$ heterogeneity.

In all experiments we measure the extent and speed of heterogeneity decrease enabled by CorDis considering $\mathcal{H}_{Disp}$ and $\mathcal{H}_{DapAvg}$. Because of space limitation we only report on $\mathcal{H}_{Disp}$, as $\mathcal{H}_{DapAvg}$ behaves the same way.

### 5.1 Impact of application parameters

In these experiments we study the impact of the volume of stored data ($r_{max}$ and $c_{max}$), the network traffic ($m_{max}$) and the initial knowledge of peers (we set different quantities of known correspondences: $init_p$). We consider five configurations (see Table 4). The configuration $c_{ref}$ serves as a reference. For these experiments we consider that fifteen different ontologies are used. Consequently $\mathcal{H}_{Rich}$ equals 0.14.

Given the alignments of reference, the heterogeneity $\mathcal{H}_{Disp}$ cannot be reduced below a certain theoric limit equal to 0.704 (cf. the solid black line on Fig. 5). This limit can be reached if the storage capacity of peers is unlimited, and if each peer $p$ knows all the $o_p$-correspondences available in the system. We anticipate that CorDis will not reduce the heterogeneity below this limit.

The graph of Fig. 5 shows that the CorDis protocol reduces $\mathcal{H}_{Disp}$ in all the configurations we set. These results allows to draw (predictable) conclusions: (i) the less peers know initially, the harder it is to reduce the heterogeneity (cf. $c_1$), (ii) the more useful information peers store, the less heterogeneous the system becomes (cf. $c_2$), and (iii) the less information peers share, the slower the heterogeneity decreases (cf. $c_4$). Nevertheless we can see that the increase of peers' cache (cf. $c_3$) does not have an important impact on heterogeneity decrease. After 50 cycles, $\mathcal{H}_{Disp}$ does not significantly vary anymore.

### 5.2 Impact of semantic richness

In these experiments we study the impact of richness heterogeneity. We vary the number of used ontologies in the system from 1 (homogenous system) to 15 (number of available ontologies in OntoFarm). As a consequence, the richness

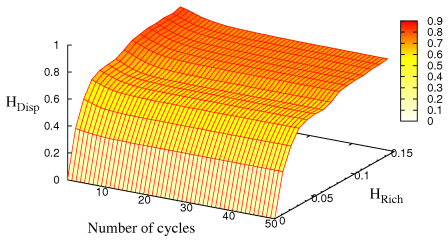


**Fig. 6.** Evolution of $\mathcal{H}_{Disp}$ in different situation of semantic richness.

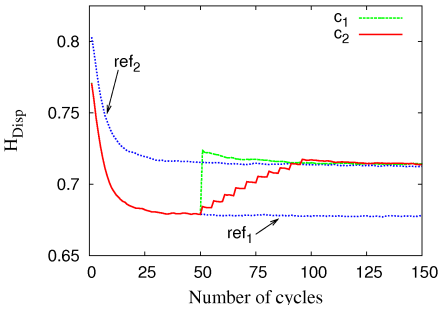


**Fig. 7.** Evolution of $\mathcal{H}_{Disp}$, and impact of peers arrival.

value $\mathcal{H}_{Rich}$ varies between 0 and 0.14. We set the other parameters as in the configuration $c_{ref}$ of the section 5.1.

Fig. 6 shows that CorDis is efficient for all the situations considered in these experiments. We plan to conduce additional experiments to show that CorDis is also efficient in highly heterogeneous systems.

### 5.3 Impact of new arrivals

In these experiments we study the impact of peers arrival in an existing system. We consider four configurations. The first one ($ref_1$), represents a system of 100 peers (using 10 different ontologies: $\mathcal{H}_{Rich} = 0.09$) in which CorDis is running. The second configuration ($ref_2$) is similar to the first one but represents a system of 110 peers (using 15 different ontologies: $\mathcal{H}_{Rich} = 0.14$). They both serve as references. In the other scenarios, 10 peers join the system simultaneously at the $50^{\text{th}}$ cycle ($c_1$) or one after the other between the $50^{\text{th}}$ cycle and the $95^{\text{th}}$ cycle: every 5 cycle a new peer joins the system ($c_2$). In both configurations, arriving peers use ontologies that are not already used, so $\mathcal{H}_{Rich}$ grows up to 0.14.

Fig. 7 shows that when a group of peers join the system, an important disruption occurs. But after 40 cycles, arriving peers are integrated in the system as if they were in it from the beginning. When peers join the system progressively, they are quickly integrated (20 cycles). As a conclusion, we can say that CorDis is robust to new arrivals.

## 6 Related work

Our measures of semantic heterogeneity assume the existence of a disparity measure between peers. Distance measures proposed in the field of ontology matching [7, 12] can be adapted, even if they do not take into account alignments between ontologies. In [5] distances between ontologies are defined in the alignment space. They can be used if we consider that queries are translated at each hop. In [3], authors define criteria to characterize the interoperability of a P2P system, but no measure is proposed to define the semantic heterogeneity of P2P systems.

CorDis aims to improve interoperability of the system by reducing some facets of the heterogeneity. Other methods have been proposed to improve interoperability. For instance in [1] authors aim to achieve a form of semantic agreement to enable queries to be forwarded to the peers that understand them best, *i.e.* with a good degree of comprehension and with correct mappings. In order to build such a system, queries are enriched with the translations used during the propagation. It enables peers to assign confidence values to the mappings. In [1, 2], the term *semantic gossiping* refers to the action of "propagating queries toward nodes for which no direct translation link exists". This is a very specific approach of gossiping which mixes both queries propagation and their translations dissemination. On the contrary our approach is independent of queries propagation and only focuses on the dissemination of correspondences. In [4] authors propose a system ensuring interoperability by offering several functionnalities to automatically organize the network of mappings at a mediation layer. Again this work can be considered as complementary to ours in the sense that the mecanism to detect the condition of strong connectivity [3] could also be put in place in the systems we consider. Others try to improve interoperability by creating a global ontology that serves as an intermediary between all peers of the system [6]. Pires *et al.* [15] present a semantic matcher which identifies correspondences between ontologies used in a PDMS. This method could be used in our context to discover correspondences, *i.e.* to initialize peers' alignments or to enrich them. In [14] authors propose to group related peers in SONs to improve interoperability. This approach is complementary to ours because they can be combined: one aims to reduce heterogeneity, and the other one aims to improve information retrieval performances.

## 7 Conclusion

With the aim of improving semantic interoparability in P2P data sharing systems we presented a new approach that consists in decreasing semantic heterogeneity. As none existed before, at least to our knowledge, we defined several measures to characterize different facets of the semantic heterogeneity of a P2P system. These measures are general enough to be used in several application domains. We proposed a new protocol, called CorDis, which relies on a gossip-based dissemination of correspondences across the system. It ensures some flexibility with respect to current queries. We conducted preliminary experiments which show that CorDis significantly reduces several facets of semantic heterogeneity. Finally, CorDis does not have any equivalent among the previously proposed solutions to improve semantic interoperability.

As future work, we first plan to conduct additional experiments with real query sets and more ontologies, as in some way, the number of ontologies limits the number of peers in the simulations. In addition, our proposal provides a basis that may be extended in several complementary directions. First, we could add a mechanism of deduction to discover new correspondences. Second, knowing correspondences might incite some peers to change their neighbourhood, thus leading to a dynamic evolution of connections. Finally, a good knowledge of

alignments between its own ontology and another one might result in a peer to adopt an additional ontology, or to change it. All these directions may help in reducing more and faster some facets of heterogeneity.

## References


[1] Karl Aberer, Philippe Cudré-Mauroux, and Manfred Hauswirth. A framework for semantic gossiping. *SIGMOD Record*, 31(4):48–53, 2002.

[2] Karl Aberer, Philippe Cudré-Mauroux, Manfred Hauswirth, and Tim Van Pelt. Gridvine: Building internet-scale semantic overlay networks. In *3rd International Semantic Web Conference*, pages 107–121, 2004.

[3] Philippe Cudré-Mauroux and Karl Aberer. A necessary condition for semantic interoperability in the large. In *3rd International Conference on Ontologies, Databases and Applications of Semantics (ODBASE)*, pages 859–872, 2004.

[4] Philippe Cudré-Mauroux, Suchit Agarwal, Adriana Budura, Parisa Haghani, and Karl Aberer. Self-organizing schema mappings in the GridVine peer data management system. In *33rd International Conference on Very Large Data Bases*, pages 1334–1337, 2007.

[5] Jérôme David, Jérôme Euzenat, and Ondřej Šváb-Zamazal. Ontology similarity in the alignment space. In *9th International Semantic Web Conference*, 2010.

[6] Herminio Camargo De Souza, Ana Maria De C. Moura, and Maria Cláudia Cavalcanti. Integrating ontologies based on P2P mappings. *IEEE Transactions on Systems, Man, and Cybernetics*, 40:1071–1082, September 2010.

[7] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, 2007.

[8] Alon Halevy, Zachary Ives, Peter Mork, and Igor Tatarinov. Piazza: data management infrastructure for semantic web applications. In *12th International World Wide Web Conference*, pages 556–567, 2003.

[9] Márk Jelasity, Rachid Guerraoui, Anne-Marie Kermarrec, and Maarten van Steen. The peer sampling service: Experimental evaluation of unstructured gossip-based implementations. In *5th International Middleware Conference*, pages 79–98, 2004.

[10] Márk Jelasity, Alberto Montresor, Gian Paolo Jesi, and Spyros Voulgaris. The Peersim simulator. `http://peersim.sf.net`.

[11] Anne-Marie Kermarrec and Maarten van Steen. Gossiping in distributed systems. *Operating Systems Review*, 41(5):2–7, 2007.

[12] Alexander Maedche and Steffen Staab. Measuring similarity between ontologies. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 251–263, 2002.

[13] Deborah L. McGuinness and Frank van Harmelen. OWL web ontology language overview. W3C recommendation, World Wide Web Consortium, 2004.

[14] Wilma Penzo, Stefano Lodi, Federica Mandreoli, Riccardo Martoglia, and Simona Sassatelli. Semantic peer, here are the neighbors you want! In *11th International Conference on Extending Database Technology*, pages 26–37, 2008.

[15] Carlos Eduardo Pires, Damires Souza, Thiago Pachêco, and Ana Carolina Salgado. A semantic-based ontology matching process for PDMS. In *2nd International Conference on Data Management in Grid and P2P Systems (Globe)*, pages 124–135, 2009.

[16] O. Šváb, V. Svátek, P. Berka, D. Rak, and P. Tomášek. Ontofarm: Towards an experimental collection of parallel ontologies. In *5th International Semantic Web Conference*, 2005. Poster Track.